Preprint

Work-Centered Support Systems: A Human-Centered Approach to Intelligent System Design

Ron Scott, Emilie Roth, Stephen Deutsch, Erica Malchiodi, Tom Kazmierczak, Robert Eggleston, Samuel Kuper and Randall Whitaker

Air Force Research Laboratory | AFRL
Science and Technology for Tomorrow's Air and Space Force

Robert G. Eggleston, PhD

Crew System Interface Division, AFRL/HECA
2255 H Street
Wright-Patterson AFB OH 45433-7022
www.hec.afrl.af.mil
Tel: (937) 255-8764 DSN: 785-8764
Fax: (937) 255-9198 E-mail: Robert.Eggleston@wpafb.af.mil

Paper cleared for public release: ASC:02-3174. Please direct all request for information or for additional copies of this report to Robert G. Eggleston at AFRL/HECA, Wright-Patterson AFB, OH, 45434

Work-Centered Support Systems: A Human-Centered Approach to Intelligent System Design

Ron Scott
BBN Technologies

Emilie M. Roth
Roth Cognitive Engineering

Stephen E. Deutsch Erika Malchiodi Tom Kazmierczak
BBN Technologies

Robert G. Eggleston Samuel R, Kuper
Air Force Research Laboratory

Randall Whitaker
Northrop Grumman Information Technology

Introduction

A hallmark of Human-Centered Computing (HCC) is that it places central focus on domain practitioners and their field of practice (Flanagan, Huang, Jones & Kasif, 1997; Hoffman, Hayes, & Ford, 2001). It takes a work-centered approach that depends on a deep analysis of the cognitive and collaborative demands of the actual field of practice, and how people work individually, in groups, and in organizations to meet those demands (Vicente, 1999; Hoffman, et al., 2002; Hoffman & Woods, 2000). The objective is to leverage what we know about human cognitive and collaborative processes so as to create systems that optimize the *affordances* (i.e., direct perception of meanings) and *effectivities* (i.e., knowledge-driven actions) for humans (Hoffman, Coffey & Ford, 2001; Flach, et al, in press).

This article describes an 'intelligent' agent-based system we have developed for a weather forecasting and monitoring application that exemplifies the HCC approach. The system was developed to demonstrate Work-Centered Support System (WCSS) concepts (Eggleston & Whitaker, 2002; Eggleston, Young & Whitaker, 2000; Young, Eggleston & Whitaker, 2000.

Much of the focus for intelligent agent technology has to date been on increased autonomy of individual software agents as well as increased coordination of activity among multiple software agents (Jennings & Wooldridge, 1996; Sycara, 1998). As the technology matures, there is a need to shift the focus to how best to deploy software agents in support of human work and how to integrate software agents into 'multi-agent' teams that include both humans and software agents (Christoffersen & Woods, in press; Malin, 2000; Malin, Johnson, Molin & Schreckenghost, 2002). The present article is intended to contribute to that dialogue.

Work-Centered Support Systems

The WCSS paradigm offers an approach for how to incorporate software agent technology in a manner that both assists the user in “keeping his or her head in the work” and reduces the possibility that the user will be surprised by software agent states or actions. This philosophy is captured in the *principle of joint aiding*: the coordinated use of direct and indirect aiding methods within a common, work-centered ontology (Eggleston et al, 2000). Indirect aiding is provided largely through the use of work domain visualizations and common work terms, appearing as labels and objects in the user interface. Direct aiding is provided by a coordinated set of software agents that interact with the user and that are clearly connected to or are embedded in the work domain visualizations.

A distinguishing characteristic of WCSS is that the focus is on supporting the multiple facets of work. The WCSS paradigm conceives of work as made up of four primary areas that require support. Specifically, a WCSS includes:
1) *Decision Support* aiding problem solving and other cognitive processes in the process of work;
2) *Product Development Support* aiding of the production of the deliverable artifact(s) of work;
3) *Collaborative Support* aiding team and colleague interactions in work, and
4) *Work Management Support* aiding the metacognitive activities entailed in prioritizing and managing multiple interweaved tasks that arise in work.

A WCSS attempts to integrate support for each of these areas through the use of direct and indirect aiding methods. By emphasizing the need to support all facets of work and stressing the need to provide both direct and indirect aiding, the WCSS paradigm falls squarely within the tradition of Human-Centered Computing.

This paper describes the weather forecasting WCSS that we developed as well as the design process that was employed, as a way of illustrating HCC. It is our contention that the building of human-centered systems requires a human-centered development process that grounds system requirements in a deep understanding of the cognitive and collaborative demands of the work domain. In our case this was achieved through close partnership between cognitive engineers, software engineers and ‘end-user’ domain practitioners—each involved from the inception of the project. The collective goal was to understand the workspace and stretch what we already knew how to do, so as to build a new work environment—a collaborative effort in envisioning, execution, and refinement of a new workspace.

The next section describes the WCSS that was developed and the human-centered design process that was used. This is followed by a general discussion that examines the principles of human-centered design and WCSS as embodied by the system. The article concludes with a reflection on agent technology and its potential to facilitate human-centered intelligent system design.

Designing Human-Centered Systems

The WCSS, called *Work-Centered Support System for Global Weather Management* (WCSS-GWM), was developed to support weather forecasting and monitoring in a military airlift service organization. Traditionally, airlift pilots have been responsible for their own flight planning, including obtaining pre-flight weather briefings. In this organization a new approach was initiated to reduce the amount of time aircrew had to devote to these tasks. A flight manager (FM) position was created with the primary responsibility for planning and managing multiple flights, both pre-flight and enroute. This includes obtaining a weather briefing and providing a complete flight plan to the pilot, including weather forecast information. The FM is viewed as a 'virtual crew member' in support of the pilot. Weather can significantly influence pre-flight and enroute flight management decisions (e.g., there may be a need to accelerate, delay or re-route a flight due to unfavorable weather conditions). As a result, weather forecasters must work closely with the FM to evaluate weather conditions at the departure and arrival airfields as well as along the planned route. The focus of our effort was on developing an intelligent system to aid near-term weather forecasting in support of planning and managing airlifts, both pre-flight and en route.

<u>Methods of Cognitive Work Analysis</u>

Consistent with the HCC philosophy, field observations and interviews intended to understand the cognitive and collaborative demands of the domain were an integral part of the WCSS-GWM development process (Roth & Patterson, in press). These revealed some aspects of cognitive and collaborative work that could benefit from decision support. These factors then guided the design of the WCSS-GWM architecture and displays.

There are many different ways to gain information about work in a domain (e.g. reading operating procedure manuals, interviewing subject matter experts, etc.) (See Hoffman, et al., 1995.) In our WCSS methodology, we emphasize acquiring an understanding of work *as practiced* to insure relevant environmental interactions are properly considered. We therefore examine work practices in terms of decision making, product development, collaborations, and we examine work management to help insure all facets of work have been reviewed.

We made a series of site visits to observe and interview flight managers and weather forecasters. The interviews and observations were conducted over the course of three site visits, each of two to three days in length, that were held approximately two months apart. Key elements of the interviews/observations included:

- Interviewing weather forecast personnel, flight managers, and senior personnel in the organization, to understand workflow and to elicit examples of situations that have occurred in the past that illustrate the kinds of complications that can arise and increase task demands;
- Observing FM and weather forecast personnel as they handled actual flights (both during planning and enroute phases) to identify additional FM/forecaster activities and sources of complexity that were not revealed by the interviews;
- Sampling as broad a range of domain practitioners and situations as practical. This included interviewing five or six individuals during each site visit, individuals of differing levels of experience and expertise, as well as conducting observations over several periods that sampled different rhythms of activity (e.g., both morning and evening shifts as well as shift turn-overs).
- Presenting to FMs for their comments a storyboard (i.e., a rapidly-developed prototype) that embodied candidate aiding concepts. The storyboard prototypes displayed increasing functionality and robustness with each site visit, reflecting what had been learned from the prior

visit. They provided a concrete vehicle to demonstrate our growing level of understanding, and obtain feedback on the viability of the evolving aiding concepts, as well as a stimulus for raising additional domain constraints and complications that needed to be accommodated.

The field observations and interviews were conducted by a multi-disciplinary team that included both cognitive engineers experienced in domain analysis and user requirements specification, as well as software engineers who would design and implement the software. The participation of both cognitive engineers and software engineers facilitated dialogue among the team members and enabled close collaboration in the specification and design of the WCSS-GWM system architecture and user-interface. At the same time, the fact that users were presented with concrete instantiations of design concepts in the form of storyboards and rapid prototypes, enabled them to participate as active partners in the design process.

The WCSS-GWM development team was able to observe the FMs and weather forecasters engaging in planning and en route monitoring of flights. This included observation of the process by which weather forecasters review and integrate multiple data sources in preparing forecasts and monitoring changes in weather, as well as the process by which they prepare products called "crew papers, which are delivered to aircrews shortly before take-off. Observations spanned both routine cases as well challenging cases. These revealed the intensive collaboration between FMs and weather forecasters that occurs when weather conditions (e.g., severe turbulence; lack of anticipated tail wind) require modification to planned flight routes (either pre-flight or en route).

Workspace and Work Patterns Observations

Typically there are three trained forecasters on duty at each shift in the organization we studied. Two sit in a weather forecast room adjacent to and within sight of the area where the FMs are located, and one sits in the flight management area to provide collaborative support to the FMs. The weather forecasting room includes a number of workstations, each with two or three CRTs. In addition there are several wall-mounted television monitors and large-screen wall-mounted displays that can be slaved to some of the workstation screens. As a matter of standard practice, the large-screen panels are set to display loops of weather satellite imagery, focused on geographic regions of interest. The large-screen panels are also used to present shift-change briefings to several FMs at once; or less frequently, they are used to support forecaster collaboration.

The focus of the forecasters was on near-term aviation weather forecasting: Understanding and predicting the weather conditions that will affect flights currently in the air as well as those scheduled to take off in the next twelve hours. Near-term aviation weather forecasting requires acquiring, interpreting and integrating multiple weather data types, including satellite imagery, observations (e.g., pilot reports of turbulence), airfield observations, upper air forecasts, and computer model projections obtained from a variety of local and worldwide sources. In practice, each of these sources is readily available to the forecaster, typically each on its own web page or map display.

In many cases, the data can be incomplete, ambiguous, stale, or conflicting. Weather forecasting expertise includes abilities to look for convergence among multiple data types (to check that 'things are lining up') as well as pursue additional sources of evidence to fill in missing data or resolve ambiguous or conflicting data. For example, a forecaster can increase confidence in a forecast by checking with a location 'upstream' in the path of a predicted storm to confirm that the storm has in fact materialized. Similarly, the forecaster can increase confidence in a prediction of turbulence by checking certain kinds of satellite imagery and/or seeking pilot reports confirming actual conditions.

We observed several cases where weather forecasters had to reconcile conflicting weather forecasts, including cases where the weather forecaster needed to call the source of a weather forecast to understand the basis for a prediction. In at least one case, the forecaster had to advocate an alternative forecast. Being able to resolve ambiguous or conflicting weather data is critical because there are risk/benefit consequences of predicting severe weather. Being too conservative can mean serious delays or cancelled missions. Being insufficiently conservative can result in costly mission diverts, damage to the plane, or even risk to life. So as to minimize impact on mission objectives, weather forecasters are very sensitive to both of these concerns and work intensely to refine their forecasts, communicate to FMs the basis and level of confidence in weather predictions, and identify ways to work around real or anticipated weather hazards (e.g., by selecting alternate take-off or landing airfields, changing the take-off time, or changing the planned flight route).

Our analysis revealed that the flight managers and weather forecasters operate closely as a team to manage the airlift missions in their work queue. The FM is the primary player, constructing flight plans, preparing crew papers, and monitoring the many details involved in ensuring the success of each mission. The forecasters engaged in multiple activities:

- Maintaining situational awareness of weather conditions in multiple geographic regions worldwide,
- Preparing general forecasts for these regions as well as tailored forecasts for each mission,
- Responding to requests and provide timely weather data to a variety of parties within the organization (including pilots who call in for weather updates),
- Monitoring weather observations to assess their impact on current and upcoming missions,
- Supporting FMs in developing options for working around hazardous weather so as to minimize impact on mission goals,
- Negotiating with other weather forecasting organizations as to the appropriate interpretation of ambiguous weather data and advocating particular weather interpretations.

Unexpected occurrences in either flight management or weather are quickly attended to by both FMs and forecasters, to better understand the effects of the changing situation and determine appropriate action (e.g., delay a flight, reroute a flight, make changes to the planned fuel load or cargo).

These findings served as the basis for defining the requirements for a WCSS.

Identifying Leverage Points

Our analysis suggested a number of opportunities to support weather forecasting and monitoring work with a WCSS.

(1). Decision Making and Product Development. These are two important elements of a complete WCSS. Forecasters can use support in collating and integrating information from multiple, disparate aviation weather sources, including forecasts, satellite imagery, real-time weather updates, and particularly flight plans for current and near-term flights). The data need to be presented within the unifying context of a geo-referenced map (cf. Hoffman, 1991). Integrating multiple weather sources on a geo-referenced map enables weather forecasters to more quickly derive and update a situation model of what significant weather factors are present in his/her geographic area of responsibility. This supports the general requirement for weather forecasters to achieve and maintain weather-related situational awareness so that they can make sound decisions. This also permits the WCSS to support weather forecasters in the cognitively challenging elements of forecasting, i.e., the identification of converging evidence in support of a forecast as well as the identification and resolution of ambiguous or conflicting information. Tools that enable more rapid recognition of changes in weather conditions would enhance both the accuracy and timeliness of forecasts. Currently, the process by which weather forecasters update/revise forecasts is labor intensive and therefore forecasts are not updated as often as forecasters would like.

(2). Support for Collaboration. Collaborative support is also needed, for the weather forecaster/FM *team* in evaluating the impact of weather on flight plans (both pre-flight and en route) and making re-route decisions.

(3). Integration of Weather and Flight Information. A third form of support involves the fact that weather information and flight information are currently not well integrated. The weather forecaster has the best access to weather information whereas the FM has the best access to the current status of planned and en route flights. Superimposing flight information (e.g., planned flight route, organized tracks that constitute legal flight path options) and weather information on a single geo-referenced map that can be simultaneously viewed by both the FM and the weather forecaster would enable the weather forecaster/FM team to directly visualize the impact of weather on a flight route and work collaboratively in formulating reroute options.

(4). Support for Work Management. Similarly, providing notification of missions that are impacted by changes in weather supports the weather forecaster and flight manager in identifying potentially high risk flights that need attention, thus providing *work management support.*

The WCSS-GWM was designed to illustrate a WCSS approach by providing all four of these kinds of support.

Using Software Agent Technology To Create A Work-Centered Support System

We began the design of the WCSS-GWM by identifying which of the top-level tasks performed by the weather forecasters in support of flight management were to be initially targeted as part of the WCSS-GWM functionality. That initial focus was on monitoring the status of en route and near-takeoff missions with respect to weather conditions, monitoring the status of certain critical regions with respect to weather conditions, and, the achievement (and maintenance of) weather-related situational awareness.

Our investigation into how weather personnel currently performed these tasks suggested which weather data types would be of use. Some of these are the same ones weather personnel already used; in some cases weather personnel evaluate graphic images and we would have to analyze the underlying numerical data to achieve the same effect. To meet our design objectives, the WCSS-GWM required the following functionalities:

- Acquisition of real-time weather observations, such as PIREPS (pilot reports) and automated observations of wind and turbulence information sent through the Aircraft Communication and Reporting System (ACARS**).**
- Acquisition of worldwide airfield and upper air forecasts produced both locally and remotely. These include SIGMETs (Significant Meteorological Information bulletins), which describe areas of weather that is potentially hazardous aviation, METARs which describe current surface weather observations at worldwide reporting stations, and terminal area forecasts (TAFs), which are bulletins which give surface weather forecasts for worldwide reporting stations.
- Graphical integration of multiple data sources – map, flight plans, forecasts, point observations, satellite imagery. An important design requirement was the ability to easily overlay any subset of data types on the same geo-referenced map for purposes of comparison.
- Automated comparison of real-time weather observations with user-defined watch areas and alerts, to focus forecaster attention on operationally relevant changes in weather conditions. While achieving a general capability for an automated alerting process would be quite difficult, we limited this to very specific (and useful) capabilities – generate an alert for any report (from PIREPS or ACARS) of turbulence or icing of at least a defined severity level in the defined region of interest (latitude, longitude, altitude, time).
- Automated and directed monitoring of individual missions and geographical areas of interest, as well as generation of alerts to focus user attention on changes in weather conditions that can impact planned or en route flights.

The WCSS-GWM prototype that was developed includes a centrally located geo-referenced, map-based visualization as a major framework for providing indirect aiding support. A variety of different weather information and flight plan information can be selectively superimposed on the map. The display is built on top of OpenMap, an open-source Java-based geospatial display toolkit (see http://openmap.bbn.com). Figure 1 shows an annotated screen shot from the WCSS-GWM that illustrates the basic features of the system.

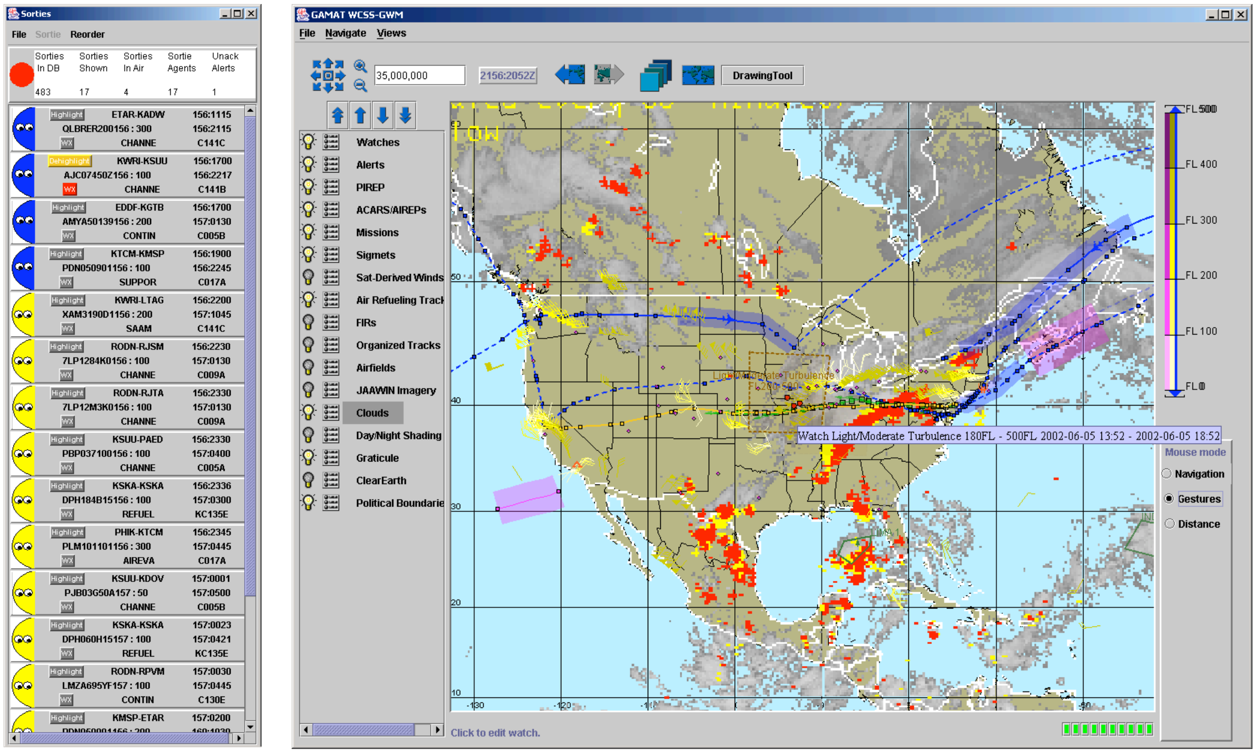

Figure 1. A screen shot from the WCSS-GWM that illustrates the basic features of the WCSS-GMW.

The "home view" of the WCSS-GWM is a map showing the geographical area of interest, with a number of controls arranged around the map. The map controls let the user pan and zoom and change projections. Layer controls allow multiple layers of flight and weather information to be viewed on the map. Flight plans, PIREPS, ACARS, observations, SIGMETs, satellite images, layers that can be placed on and taken off from the map. An altitude slider control allows the user to filter observations by specifying an altitude area of interest. More details can be accessed by hovering over an icon. For example, the text of a PIREP can be obtained by placing the mouse over the PIREP symbol displayed on the map. Thus, both the weather information that is included as layers and the labels that are used to index them reflect the first-person, work-centered philosophy of HCC.

This philosophy is extended to a floating Sortie Palette that provides an overall summary of all missions of interest, status of individual missions, the ability to highlight and locate specific missions, and the ability to sort and organize them to suit the work context. It also enables users to maintain awareness of weather-related alerts and keep track of which have already been viewed and which remain to be dealt with. Further, it is integrated with the map display in that problem notifications can be directly highlighted on the map with a simple button click on the palette. Thus, the Sortie Palette aids work management both for work on a given mission and for working with a set of missions.

The Role of Software Agents

Software agents provide intelligent automation to directly support to the forecaster. WCSS-GWM includes agents that monitor missions and watch areas, and notify the forecaster when operationally significant changes in weather occur. A key design element of the WCSS-GWM is that these agents can be created, monitored, and modified by the forecaster. For example, the forecaster can create an agent by drawing a polygon around a geographic region of interest (a watch area) on the map and specifying the agent behavior (desired altitude, start and stop time, hazard type and severity to watch for). At a later time, the forecasterer can modify the agent behavior by changing these parameters, as well as modify the shape and position of the polygon. Forecasters can also create and modify agents that monitor for changes in weather around a flight path.

Figure 2 shows a screen shot from the WCSS-GWM that illustrates the ability to create and modify agents. The wide blue shading along a flight path indicates the geographic area along a flight path that is being monitored by an agent. Similarly, the transparent geometric shapes (off the Eastern US coastline) provide visual indication of the watch areas being monitored by agents. The 'create watch area' pop-up window and 'edit agent parameters' pop-up window illustrate the ability to create new agents, and to view and modify parameters that control agent behavior.

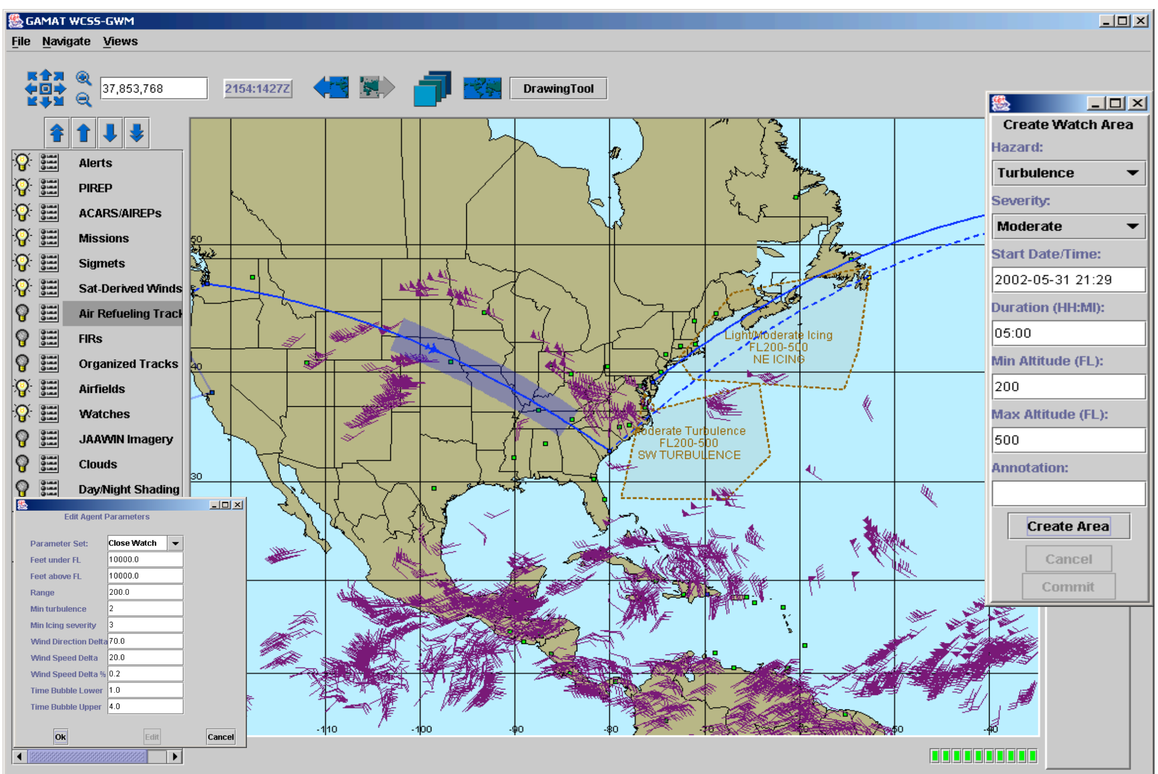

Figure 2. A screen shot from the WCSS-GWM that illustrates the ability to create and modify software agents.

We regard the *user* (forecaster) creation of agents as an important aspect of a WCSS. It is well known that complex systems that have many automation features, such as auto assistance in different operational modes, can lead to machine-induced user errors (Christofferson & Woods, in press), in which the user thinks the system is behaving in one way while the automation condition causes the system to behave differently. This problem can be mitigated by having the user create and set the conditions for a software agent. Because of this involvement in specifying the automation protocol and activating it, the user should have increased awareness of what service the automation is providing.

It is also important to note that the watch area agent shown in Figure 2 is expressed in work terms (a physical area to be watched- indicated by the shaped polygon). This means the user does not have to translate from an agent icon to the weather-based semantic intent of what assistance the agent is providing. The agent is observed directly in the ontology of work, thereby reducing cognitive complexity and demand.

Our WCSS-GWM contains three broad classes of agents:

(1). Acquisition Agents acquire data from outside sources (e.g., weather bulletins, ACARS, SIGMETs, satellite imagery, mission details, flight plans). Each acquisition agent is responsible for a particular data type/source, and will periodically retrieve the latest data from that source (anywhere from once a minute to once every few hours, depending on how often new data is available from the source). Furthermore, each acquisition agent signals other interested agents when new data have been retrieved.

(2). Analysis Agents analyze data retrieved by acquisition agents to produce initial problem indications (individual turbulence reports, lightning strike reports, intersections of flight plans with SIGMES, etc.) Types of analysis agents include: *Region analysis agents* that are triggered by the weather forecasters when they decide to monitor a geographic region for critical conditions (i.e., create a watch area), and then watch for observations matching given criteria; and *Mission analysis agents* that are automatically generated by the presence of a current or upcoming flight mission. This agent watches for reports (e.g., PIREPS or ACARS) close to the flight plan (in latitude, longitude, altitude, time space) that significantly affect the mission.

(3). Presentation Agents are based on the results of the analysis agents, and decide what information is presented to the user. These agents work on initial problem indications, clustering and prioritizing, to present high-level presentation of problems. For example, there may be many related notifications generated by the analysis agents that need to be aggregated together into a single notification message to avoid an 'alarm avalanche' problem (Woods, 1995). Presentation agents are also responsible for staging displays, that is, retrieving enough data, and the right kind of data, so that the information needed by the user can be quickly rendered on the screen. We have implemented only a limited presentation agent capability (as shown in Figure 2), but in a full-scale implementation of a WCSS-GWM these agents would have two additional responsibilities:

- Displaying data at different levels of aggregation, depending on the user's role. For example, a supervisor may get only a top-level summary view of areas of turbulence, while the user responsible for a particular mission might see individual reports of turbulence close to that mission path. It would be the job of the presentation agent to aggregate the same underlying data to different levels for the different users.
- Displaying different data, depending on the user's role. In a global system, multiple flight managers and weather forecasters would split the globe into regions of responsibility. As indications of critical weather are produced by the analysis agents, it would be the job of a presentation agent to present this information to only those users who would be interested.

One of the considerations in the design of the agent architecture was to create a structure that could be understandable, inspectable, and modifiable by the forecasters. While the literature on software agents has tended to focus on the high level tasks delegated to agents and their level of autonomy, the value of agent technology from a software development perspective is that software agents are small, independent 'chunks' of software that each address a small unified set of tasks, are separately controllable, and separately modifiable. In creating the agent architecture, a key consideration was to structure the software so that the capabilities of the software 'chunks' implemented as agents will be meaningful to the user in terms of his/her work domain, as indicated earlier. The agents are configured in such a manner as to mirror the basic terms of reference the user employs in addressing his/her work. This applies both with respect to the agents' functions (e.g., acquiring, analyzing, and presenting data) and to the domain objects that the software agents work on (e.g., missions, forecasts, watch areas). Once the software is organized into domain meaningful 'chunks' implemented as software agents, users can more readily observe and direct their operation.

Using D-OMAR to Create Agent-based Systems

Our agent system is built on top of the D-OMAR (Distributed Object Model Architecture) system developed under the sponsorship of the Air Force Research Laboratory (see http://omar.bbn.com). D-OMAR was initially designed as a set of languages and tools to support the development of knowledge-based simulations of human performance, with a focus on the cognitive skills of the human operator, and has evolved into a full-featured distributed agent infrastructure. It provides a set of services that is essential to agent-based system development. Agents must be created, have the ability to launch procedures that implement their services, and retire or be removed when no longer needed. A publish-subscribe protocol supports communication among agents. In addition to the basic function of moving data between agents, the publish-subscribe capability plays an essential role in coordinating the activities of pairs or groups of agents. Furthermore, the publish-subscribe protocol used for inter-agent communication can also be used by an agent to coordinate the execution of its multiple proactive and reactive behaviors. We have found these language features to be essential to the rapid development of sophisticated agent behaviors.

There exist two implementations of D-OMAR: the original Lisp implementation now known as OmarL, and the new Java implementation, OmarJ. WCSS-GWM development was done using OmarJ. The software agents developed in either environment have similar capabilities; indeed, the interoperability of Java-based and Lisp-based agents is supported. The implementation of agent behaviors is supported through the provision of a procedural language. In Lisp, the Simulation Core language (SCORE) is provided as an extension of Lisp itself. ScoreJ, a set of Java class libraries, provides a similar capability for Java implementations. Important language features (e.g., *race, join, satisfy*) support the development of concurrent proactive and reactive procedures essential to individual agent behaviors. Time management (e.g., *sleep*) is provided to support the scheduling of important services and may be used with concurrency control to specify "time out" conditions. These language features used in combination may act as pattern matchers for events evolving in time. The languages also support the priority-based mediation of contention between specified subsets of procedures. The publish-subscribe protocol used for inter-agent communication can also be used by an agent to coordinate the execution of its multiple proactive and reactive behaviors. We have found these language features to be essential to the rapid development of sophisticated agent behaviors.

Current Status of WCSS-GWM

The WCSS-GWM is currently in the late stage of development. It has been installed on several workstations within the airlift service facility and is being used for forecasters. There have been several reports of cases where flights were successfully rerouted based on information provided by the WCSS-GWM.

General Discussion

Hoffman et al. (2002) have argued that what distinguishes HCC is that it takes a particular stance on the design, development, and evaluation of technology that enhances human cognitive and collaborative activities. The WCSS-GWM embodies this approach. Work-centered support is provided directly and indirectly through the integration of work-centered visualizations and focused and localized software agents.

We argue, and try to show by example, that human-centered systems entail a human-centered design process so as to insure a human-centered product. In the case of the WCSS-GWM, the system requirements were established by an interdisciplinary team with a deep commitment to performing an analysis of work patterns as a basis for defining system requirements. The system requirements were driven by an analysis of the sources of cognitive and collaborative demands in the field of practice, spanning the four components of work as defined by the WCSS methodology. It involved close interaction among the cognitive engineers, software developers and domain practitioners.

A growing body of cognitive engineering literature has shown that in order for automated agents to be effective they must act as ‘team players’ (Roth, Malin & Schreckenghost, 1997; Malin, 2000; Christoffersen & Woods, in press). For software agents to become team players, there are two fundamental characteristics that need to be designed in from the beginning—*observability* and *directability*. Users need to be able to ‘see’ what the automated agents are doing and understand what they will do next relative to the state of the task. Humans also need to be able to control and re-direct the software agents as task requirements change. The WCSS-GWM agent-based architecture was designed with these objectives in mind. The geo-referenced map with weather and flight information superimposed provides a representation "common ground" representation of the current situation that is available to the humans (the FMs and the weather forecasters) and the software agents that are involved in interpreting weather and its implications for flight missions. Furthermore, the activities of the agents are directly visible and controllable by the users—the geographic area being monitored by the software agents (both with regard to a flight mission and with regard to forecast and watch areas) is explicitly presented on the display and can be modified by the user. Similarly, the weather parameters that are being monitored by the agents, and the trigger points for alerts can be inspected and modified by the forecaster.

The WCSS brings an additional idea to the design of decision-aiding software, one that, we believe, extends the meaning of human-centered. In particular, it emphasizes that for a system to be truly human-centered, it needs to support the multiple facets of individual cognitive and collaborative work. This includes consideration of not only the problem-solving/decision-making aspect of work, but also the activities involved in creation of work products, the processes entailed in collaborative work, and the cognitive effort involved in the tracking and management of multiple intertwined work activities (e.g., requirements for attention shift and memory of the number and state of tasks in process).

Reflections on the Role of Software Agents in Human-Centered Systems

Our discussion of agents in the WCSS-GWM points to multiple 'roles' of agents in the development of human-centered systems. At the software development level, the value of agent technology is that the software agents are small, independent 'chunks' of software that each address bounded tasks that are separately controllable and separately modifiable. This greatly facilitates software development, maintenance and upgrade.

From a work-centered perspective, the utility of software agents is that they enable software components to be organized around the functional elements of work in the domain. A detailed domain analysis allows the development team to map out the domain work requirements and systematically allocate the discerned tasks to human and software agents as appropriate. In the domain of weather forecasting in support of flight management, the kinds of work that can be allocated to software agents include 'watch this region of the world and let me know of any weather problems that arise' (i.e., region analysis agents); 'watch this mission for me and let me know if any problems that arise' (i.e., mission analysis agents) or even 'watch this weather system' or 'interpret this weather pattern' (i.e., weather-centered agents – yet to be implemented), freeing human attention resources for other tasks.

A corollary of the structuring of agents in functional terms is that the agents give us a concrete vocabulary of concepts and metaphors (agents as subordinates to whom you can assign specific tasks) that can be shared by software engineers, cognitive engineers and users. This facilitates communication and enables closer collaboration during the design phase of a system.

Software Agents: Behind the Scenes vs. Up-Front Presence

Closely related to the question of the role of software agents in human-centered systems, is the question of the 'visibility' of the software agents in the interface. A question we continue to struggle with is the extent to which (and conditions under which) software agents should have an explicit screen presence. Should software agents be treated as convenient 'chunks of code' that underlie system functionality but have no explicit presence in the user interface, or should they be explicit in the user interface so as to allow users to knowingly interact with them?

There are multiple possible stances on this question. At one extreme it might be argued that agents are a powerful tool for software implementation, but that users should never see (or hear about) them as such. Arguments for this position include concerns that users may ascribe a greater degree of expertise, competence and autonomy to the agents than is warranted. Such ascriptions could lead users to overly 'trust' the software agents. Conversely, it could also cause users to unjustifiably 'mistrust' or even 'fear' the agents (i.e., ascribing a greater level of autonomy than is warranted). At the other extreme, one might argue that providing the agents with a screen presence and even a 'personality' (e.g., the animated Microsoft paperclip) can allow users to more readily grasp the functionality inherent in the software agents by taking advantage of metaphors such as 'assistant' or 'subordinate' to whom one can delegate tasks.

For the WCSS-GWM we chose a middle ground. Some of the agents are visible in the user interface, while others operate behind the scenes. The agents that have a visible presence in the user interface are the agents that are organized around domain work: The forecast analysis agents, region analysis agents and mission analysis agents. These are the agents that users 'delegate' work to that they would otherwise have to do themselves. The presence of these agents in the user interface enables the users to monitor and control their performance. Our focus is on allowing users to understand, observe and control the behavior of the agents. However, we made no attempt to personify these agents. There is nothing that marks them as 'agents' on the screen. They have no 'personality' or animated presence. Rather, we adopted a work-centered perspective where users interact with the graphical user interface to achieve work and monitor and 'task' agents in the context of doing work. Other agents in the WCSS-GWM remain predominantly behind the scenes but users can access them should the need arise. Examples of this are the data acquisition agents. Although most of the time these agents operate out of sight, forecasters can bring up displays allowing inspection and modification of their performance as necessary. This becomes particularly important for diagnosing agent behaviors (e.g., if a data source accessible over the web goes down; if data become corrupted or stale) or tailoring agent behaviors to new circumstances (e.g., to add a new data source or modify a data source).

Finally, there are still other agents that execute software task elements but are not visible or controllable by the users. This category includes the presentation agents, where a need for user interaction with the agents has not (yet) emerged.

<u>On Adding Agent-based Systems to the Mix of End Users, Cognitive Engineers, and System Developers</u>

We found that an additional benefit of software agent technology is its potential to facilitate collaboration between users, the cognitive engineers and the Software Engineers. The adoption of object-oriented design techniques over the last ten or fifteen years improved communication between software engineers, cognitive engineers and end users by providing easily understandable descriptions of the software implementation of domain objects and small-scale behavior of those objects. These descriptions have at times fallen short, though, in providing the user with a clear understanding of the large-scale system behavior—how objects hook together to produce the desired system results. As a system increases in complexity, the user can easily lose the big picture of how objects work together, with the result that collaboration the user and the software development team effectively stops.

The agent architecture provides the potential for user-accessible descriptions of not only domain objects, but also of workflow and large-scale interactions between domain objects. The design of the software is stated in terms that all members of the team can understand and contribute to. Our expectation is that the dialog among users, cognitive engineers, and software engineers couched in terms of workflow for users and software agents will contribute to improved human-centered system design.

References

Christoffersen, K. & Woods, D. D. (in press). How to make automated systems team players. In E. Salas, (Ed.), Advances in human performance and cognitive engineering research, Vol. 2 (pp. ###-###). JAI Press/Elsevier.

Eggleston, R. G. and Whitaker, R. D. (2002). Work-Centered Support Systems Design: Using Organizing Frames to Reduce Work Complexity. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting* (pp. 265-269). Santa Monica, CA: Human Factors and Ergonomics Society.

Eggleston, R.G., Young, M. J. & Whitaker, R. D. (2000, October). Work-centered support system technology: a new interface client technology for the Battlespace Infosphere. *Proceedings of NEACON 2000* (pp. 499-506). City: Publisher.

Flach, J.M., Vicente, K.J. , Tanabe, F., Monta, K. & Rasmussen, J. (In press). Ecological approach to interface design. In K. Itoh & S. Kuwano (Eds.) *Handbook of human factors/ergonomics* (pp. ###-###). Tokyo: Asakura Publishing Co.

Flanagan, J., Huang, T., Jones, P. & Kasif, S. (1997). *Human-Centered systems: Information, interactivity, and intelligence*. Washington, DC: National Science Foundation.

Hoffman, R. R. (1991). Human factors psychology in the support of forecasting: The design of advanced meteorological workstation. *Weather & Forecasting, ##,* 98-110.

Hoffman, R. R., Coffey, J. W. & Ford, K. M. (2001). The handbook of human-centered computing. Institute for Human and Machine Cognition, University of West Florida Technical Report.

Hoffman, R. R., Feltovich, P. J., Ford, K. M., Woods, D. D., Klein, G., & Feltovich, A. (July/August 2002). A rose by any other name… would probably be given an acronym. *IEEE: Intelligent Systems*, 72-80.

Hoffman, R. R., Hayes, P. J., & Ford, K. M. (2001/September-October). Human-Centered computing: Thinking in and outside the box. *IEEE: Intelligent Systems*, pp. 76-78.

Hoffman R. R. & Woods. D. D. (2000). Special Section on Cognitive Task Analysis. *Human Factors*, 42(1).

Jennings, N. & Wooldridge, M. (1996, January). Software agents. *IEEE Review,* pp. 17-20.

Malin, J. (2000). Preparing for the unexpected: Making remote autonomous agents capable of interdependent teamwork. *Proceedings of the XIVth Triennial Congress of the International Ergonomics ssociation and 44th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 1-254 - 1-257). Santa Monica, CA: Human Factors and Ergonomics Society.

Malin, J. T., Johnson, K., Molin, A., & Schreckenghost, D. (2002, March). Integrated tools for mission operations teams and software agents**.** *IEEE Aerospace Conference Proceedings* (pp. ###-###). City, State: IEEE.

Roth, E. M. and Patterson, E. S. (in press). Using observational study as a tool for discovery: Uncovering cognitive and collaborative demands and adaptive strategies**.** In B. Brehmer, R. Lipshitz, R., & H. Montgomery, (Eds.). *How do professionals make decisions?* (pp. ###-###). Mahaw, NJ: Erlbaum.

Roth, E. M., Patterson, E. S. & Mumaw, R. J. (2002). Cognitive engineering: Issues in user-centered system design. In J. J. Marciniak (Ed.), *Encyclopedia of software engineering (2nd ed.)* (pp. 163–179). New York: Wiley-Interscience.

Roth, E., Malin, J., & Schreckenghost, D. (1997). Paradigms for intelligent interface design. In M. Helander, T. Landauer, & P. Prabhu (Eds.), *The Handbook of human-computer interaction (2nd ed.)* (pp. 1177-1201). Amsterdam: Elsevier.

Sycara, K. (1998). Multiagent systems. *AI Magazine*, *19*(2), 79-92.

Vicente, K. J. (1999). *Cognitive work analysis: Towards safe, productive, and healthy computer-based work*. Mahwah, NJ: Erlbaum.

Woods, D. D. (1995). The alarm problem and directed attention in dynamic fault management. *Ergonomics*, 38, 2371-2393.

Young, M.J., Eggleston, R. G., and Whitaker, R. D. (2000, April). Direct manipulation interface techniques for interaction with software agents. Paper presented at the *NATO/RTO Symposium on Usability of Information in Battle Management Operations*, Oslo, Norway.